# Red Hat Enterprise Linux OpenStack Platform 4
# Release Notes

---

Release Notes for Red Hat Enterprise Linux OpenStack Platform 4 (Havana)

11 Nov 2014

Red Hat Engineering Content Services

# Red Hat Enterprise Linux OpenStack Platform 4 Release Notes

## Release Notes for Red Hat Enterprise Linux OpenStack Platform 4 (Havana)

11 Nov 2014

Red Hat Engineering Content Services

## Legal Notice

Copyright © 2013-2014 Red Hat, Inc.

This document is licensed by Red Hat under the Creative Commons Attribution-ShareAlike 3.0 Unported License. If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, MetaMatrix, Fedora, the Infinity Logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack Logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## Abstract

The Release Notes document the major features, enhancements, and known issues of the Red Hat Enterprise Linux OpenStack Platform 4 (Havana) release.

# Table of Contents

**Preface** ..... **2**
1. Document Conventions 2
1.1. Typographic Conventions 2
1.2. Pull-quote Conventions 3
1.3. Notes and Warnings 4
2. Getting Help and Giving Feedback 4
2.1. Do You Need Help? 4
2.2. We Need Feedback! 5

**Chapter 1. Product Introduction** ..... **6**
1.1. About this Release 6
1.2. Operating System Requirements 7
1.3. RHN/CDN Channels 7
1.3.1. If Using CDN 7
1.3.2. If Using RHN 8
1.3.3. Package Notes 9
1.4. Product Support 9
1.5. Additional Documentation 9

**Chapter 2. Release Notes** ..... **11**
2.1. Enhancements 11
2.2. Technology Preview 37
2.3. Recommended Practices 38
2.4. Known Issues 40
2.5. Deprecated Functionality 47

**Chapter 3. Upgrading** ..... **48**
3.1. Upgrade Overview 48
3.2. Method - Upgrade All at Once 50
3.3. Method - Upgrade Service-by-Service 52
3.3.1. Upgrade an individual service 52
3.3.2. Finalize system upgrade 55
3.4. Method - Upgrade Service-by-Service with Parallel Computes 56
3.4.1. Set up a parallel Compute environment 56
3.4.2. Move instances to the new environment 60

**Revision History** ..... **61**

# Preface

## 1. Document Conventions

This manual uses several conventions to highlight certain words and phrases and draw attention to specific pieces of information.

### 1.1. Typographic Conventions

Four typographic conventions are used to call attention to specific words and phrases. These conventions, and the circumstances they apply to, are as follows.

**`Mono-spaced Bold`**

Used to highlight system input, including shell commands, file names and paths. Also used to highlight keys and key combinations. For example:

> To see the contents of the file **`my_next_bestselling_novel`** in your current working directory, enter the **`cat my_next_bestselling_novel`** command at the shell prompt and press **`Enter`** to execute the command.

The above includes a file name, a shell command and a key, all presented in mono-spaced bold and all distinguishable thanks to context.

Key combinations can be distinguished from an individual key by the plus sign that connects each part of a key combination. For example:

> Press **`Enter`** to execute the command.
>
> Press **`Ctrl`**+**`Alt`**+**`F2`** to switch to a virtual terminal.

The first example highlights a particular key to press. The second example highlights a key combination: a set of three keys pressed simultaneously.

If source code is discussed, class names, methods, functions, variable names and returned values mentioned within a paragraph will be presented as above, in **`mono-spaced bold`**. For example:

> File-related classes include **`filesystem`** for file systems, **`file`** for files, and **`dir`** for directories. Each class has its own associated set of permissions.

**Proportional Bold**

This denotes words or phrases encountered on a system, including application names; dialog-box text; labeled buttons; check-box and radio-button labels; menu titles and submenu titles. For example:

> Choose **System** → **Preferences** → **Mouse** from the main menu bar to launch **Mouse Preferences**. In the **Buttons** tab, select the **`Left-handed mouse`** check box and click **`Close`** to switch the primary mouse button from the left to the right (making the mouse suitable for use in the left hand).
>
> To insert a special character into a **`gedit`** file, choose **Applications** → **Accessories** → **Character Map** from the main menu bar. Next, choose **Search** → **Find…** from the **Character Map** menu bar, type the name of the character in the **Search** field and click **`Next`**. The character you sought will be highlighted in the

> **Character Table**. Double-click this highlighted character to place it in the **Text to copy** field and then click the **Copy** button. Now switch back to your document and choose **Edit** → **Paste** from the **gedit** menu bar.

The above text includes application names; system-wide menu names and items; application-specific menu names; and buttons and text found within a GUI interface, all presented in proportional bold and all distinguishable by context.

***Mono-spaced Bold Italic*** or ***Proportional Bold Italic***

Whether mono-spaced bold or proportional bold, the addition of italics indicates replaceable or variable text. Italics denotes text you do not input literally or displayed text that changes depending on circumstance. For example:

> To connect to a remote machine using ssh, type **ssh *username*@*domain.name*** at a shell prompt. If the remote machine is **example.com** and your username on that machine is john, type **ssh john@example.com**.
>
> The **mount -o remount *file-system*** command remounts the named file system. For example, to remount the **/home** file system, the command is **mount -o remount /home**.
>
> To see the version of a currently installed package, use the **rpm -q *package*** command. It will return a result as follows: ***package-version-release***.

Note the words in bold italics above: username, domain.name, file-system, package, version and release. Each word is a placeholder, either for text you enter when issuing a command or for text displayed by the system.

Aside from standard usage for presenting the title of a work, italics denotes the first use of a new and important term. For example:

> Publican is a *DocBook* publishing system.

## 1.2. Pull-quote Conventions

Terminal output and source code listings are set off visually from the surrounding text.

Output sent to a terminal is set in **mono-spaced roman** and presented thus:

```
books        Desktop   documentation  drafts  mss    photos   stuff  svn
books_tests  Desktop1  downloads      images  notes  scripts  svgs
```

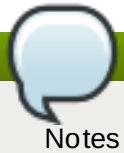
Source-code listings are also set in **mono-spaced roman** but add syntax highlighting as follows:

```
static int kvm_vm_ioctl_deassign_device(struct kvm *kvm,
                  struct kvm_assigned_pci_dev *assigned_dev)
{
         int r = 0;
         struct kvm_assigned_dev_kernel *match;

         mutex_lock(&kvm->lock);

         match = kvm_find_assigned_dev(&kvm->arch.assigned_dev_head,
                                       assigned_dev->assigned_dev_id);
         if (!match) {
                 printk(KERN_INFO "%s: device hasn't been assigned
```

```
before, "
                        "so cannot be deassigned\n", __func__);
                r = -EINVAL;
                goto out;
        }

        kvm_deassign_device(kvm, match);

        kvm_free_assigned_device(kvm, match);

out:
        mutex_unlock(&kvm->lock);
        return r;
}
```

### 1.3. Notes and Warnings

Finally, we use three visual styles to draw attention to information that might otherwise be overlooked.



**Note**

Notes are tips, shortcuts or alternative approaches to the task at hand. Ignoring a note should have no negative consequences, but you might miss out on a trick that makes your life easier.

**Important**

Important boxes detail things that are easily missed: configuration changes that only apply to the current session, or services that need restarting before an update will apply. Ignoring a box labeled “Important” will not cause data loss but may cause irritation and frustration.

**Warning**

Warnings should not be ignored. Ignoring warnings will most likely cause data loss.

## 2. Getting Help and Giving Feedback

### 2.1. Do You Need Help?

If you experience difficulty with a procedure described in this documentation, visit the Red Hat Customer Portal at http://access.redhat.com. Through the customer portal, you can:

- search or browse through a knowledgebase of technical support articles about Red Hat products.
- submit a support case to Red Hat Global Support Services (GSS).
- access other product documentation.

Red Hat also hosts a large number of electronic mailing lists for discussion of Red Hat software and technology. You can find a list of publicly available mailing lists at https://www.redhat.com/mailman/listinfo. Click on the name of any mailing list to subscribe to that list or to access the list archives.

## 2.2. We Need Feedback!

If you find a typographical error in this manual, or if you have thought of a way to make this manual better, we would love to hear from you! Please submit a report in Bugzilla: http://bugzilla.redhat.com/ against the product **Red Hat OpenStack.**

When submitting a bug report, be sure to mention the manual's identifier: *doc-Release_Notes*

If you have a suggestion for improving the documentation, try to be as specific as possible when describing it. If you have found an error, please include the section number and some of the surrounding text so we can find it easily.

# Chapter 1. Product Introduction

Red Hat Enterprise Linux OpenStack Platform provides the foundation to build a private or public Infrastructure-as-a-Service (IaaS) cloud on top of Red Hat Enterprise Linux. It offers a massively scalable, fault-tolerant platform for the development of cloud-enabled workloads.

Red Hat Enterprise Linux OpenStack Platform is packaged so that available physical hardware can be turned into a private, public, or hybrid cloud platform including:

- Fully distributed object storage
- Persistent block-level storage
- Virtual-machine provisioning engine and image storage
- Authentication and authorization mechanism
- Integrated networking
- Web browser-based GUI for both users and administration.

The Red Hat Enterprise Linux OpenStack Platform IaaS cloud is implemented by a collection of interacting services that control its computing, storage, and networking resources. The cloud is managed using a web-based interface which allows administrators to control, provision, and automate OpenStack resources. Additionally, the OpenStack infrastructure is facilitated through an extensive API, which is also available to end users of the cloud.

For detailed information about Red Hat Enterprise Linux OpenStack Platform, refer to the books listed in Section 1.5, “Additional Documentation”.

## 1.1. About this Release

This release of Red Hat Enterprise Linux OpenStack Platform is based on the OpenStack "Havana" release. It includes additional features, known issues, and resolved issues specific to Red Hat Enterprise Linux OpenStack Platform.

Only changes specific to Red Hat Enterprise Linux OpenStack Platform are included in this release notes document. The release notes for the OpenStack "Havana" release itself are available at this location:

**OpenStack "Havana" Release Notes**

https://wiki.openstack.org/wiki/ReleaseNotes/Havana

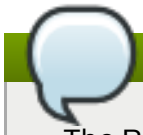
**Note**

- Red Hat Enterprise Linux OpenStack Platform uses components from other Red Hat products. Specific information pertaining to the support of these components is available at:

  https://access.redhat.com/site/support/policy/updates/openstack/platform/

- To evaluate Red Hat Enterprise Linux OpenStack Platform, sign up at:

  http://www.redhat.com/openstack/.



**Note**

The Red Hat Enterprise Linux High Availability Add-On is available for Red Hat Enterprise Linux OpenStack Platform use cases. See the following URL for more details on the add-on: http://www.redhat.com/products/enterprise-linux-add-ons/high-availability/. See the following URL for details on the package versions to use in combination with Red Hat Enterprise Linux OpenStack Platform: https://access.redhat.com/site/solutions/509783

## 1.2. Operating System Requirements

Red Hat Enterprise Linux OpenStack Platform 4 requires Red Hat Enterprise Linux 6.6 Server (v6.4 is no longer supported).

For further information on installing Red Hat Enterprise Linux 6.6 Server, refer to the *Red Hat Enterprise Linux 6 Installation Guide*.

## 1.3. RHN/CDN Channels

The following channels are required to access Red Hat Enterprise Linux OpenStack Platform 4.

**Warning**

Although older Red Hat OpenStack repositories are available, you must ensure that your system can no longer access them before installing Red Hat Enterprise Linux OpenStack Platform 4. For example, for CDN, unsubscribe from or disable the following:

- Red Hat OpenStack 1.0 (Essex) -- rhel-server-ost-6-preview-rpms
- Red Hat OpenStack 2.1 (Folsom) -- rhel-server-ost-6-folsom-rpms
- Red Hat Enterprise Linux OpenStack Platform 3 (Grizzly) -- rhel-server-ost-6-3-rpms
- Red Hat Enterprise Linux OpenStack Platform 4 Beta (Havana) -- rhel-6-server-openstack-beta-rpms

### 1.3.1. If Using CDN

Content Delivery Network (CDN) channels can be:

- Enabled with the **`subscription-manager repos --enable=[reponame]`** command.
- Disabled with the **`subscription-manager repos --disable=[reponame]`** command.

**Table 1.1. Required Channels**

| Channel | Repository Name |
|---|---|
| Red Hat OpenStack 4.0 (RPMS) | **`rhel-6-server-openstack-4.0-rpms`** |
| Red Hat Enterprise Linux 6 Server (RPMS) | **`rhel-6-server-rpms`** |

**Table 1.2. Optional Channels**

| Channel | Repository Name |
|---|---|
| RHEL Server Load Balancer (v6 for 64-bit x86_64) | **`rhel-lb-for-rhel-6-server-rpms`** |

If using CDN, the following channels must be disabled in order for Red Hat Enterprise Linux OpenStack Platform 4 to function correctly.

**Table 1.3. Disable Channels**

| Channel | Repository Name |
|---|---|
| Red Hat CloudForms Management Engine | **`"cf-me-*"`** |
| Red Hat CloudForms Tools for RHEL 6 | **`"rhel-6-server-cf-*"`** |
| Red Hat Enterprise Virtualization | **`"rhel-6-server-rhev*"`** |
| Red Hat Enterprise Linux 6 Server - Extended Update Support | **`"*-eus-rpms"`** |

### 1.3.2. If Using RHN

RHN software channels can be:

- Added using the **`rhn-channel --add --channel=[reponame]`** command.
- Removed using the **`rhn-channel --remove --channel=[reponame]`** command.

**Table 1.4. Required Channels**

| Channel | Repository Name |
|---|---|
| Red Hat OpenStack 4.0 (for RHEL 6 Server x86_64) | **`rhel-x86_64-server-6-ost-4`** |
| Red Hat Enterprise Linux Server (v6 for 64-bit AMD64 / Intel64) | **`rhel-x86_64-server-6`** |
| MRG Messaging v2 (for RHEL 6 Server x86_64) | **`rhel-x86_64-server-6-mrg-messaging-2`** |

**Table 1.5. Optional Channels**

| Channel | Repository Name |
|---|---|
| RHEL Server Load Balancer (v6 for 64-bit x86_64) | **`rhel-x86_64-server-lb-6`** |

### 1.3.3. Package Notes

- The Red Hat Common for RHEL Server (v6) channel is recommended for use if creating custom Red Hat Enterprise Linux guest images that require cloud-init.

```
# subscription-manager repos \
    --enable=rhel-6-server-rh-common-rpms
```

## 1.4. Product Support

Available resources include:

**Customer Portal**

The Red Hat Customer Portal offers a wide range of resources to help guide you through planning, deploying, and maintaining your OpenStack deployment. Facilities available via the Customer Portal include:

- Knowledge base articles and solutions.
- Reference architectures.
- Technical briefs.
- Product documentation.
- Support case management.

Access the Customer Portal at https://access.redhat.com/.

**Mailing Lists**

Red Hat provides these public mailing lists that are relevant to OpenStack users:

- The **rhsa-announce** mailing list provides notification of the release of security fixes for all Red Hat products, including Red Hat Enterprise Linux OpenStack Platform.

  Subscribe at https://www.redhat.com/mailman/listinfo/rhsa-announce.

- The **rhos-list** mailing list provides a forum for discussions about installing, running, and using OpenStack on Red Hat based distributions.

  Subscribe at https://www.redhat.com/mailman/listinfo/rhos-list.

**Note**

The full list of updates released for Red Hat Enterprise Linux OpenStack Platform is maintained at https://rhn.redhat.com/errata/rhel6-rhos-4-errata.html.

## 1.5. Additional Documentation

You can find additional documentation for the Red Hat Enterprise Linux OpenStack Platform in the customer portal:

https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform

The following documents are included in the documentation suite:

**Table 1.6. Red Hat Enterprise Linux OpenStack Platform documentation suite**

| Guide | Description |
|---|---|
| *Administration User Guide* | HowTo procedures for administrating Red Hat Enterprise Linux OpenStack Platform environments. |
| *Configuration Reference Guide* | Configuration options and sample configuration files for each OpenStack component. |
| *End User Guide* | HowTo procedures for using Red Hat Enterprise Linux OpenStack Platform environments. |
| *Getting Started Guide* | PackStack deployment procedures for a Red Hat Enterprise Linux OpenStack Platform cloud, as well as brief How-to's for getting your cloud up and running. |
| *Installation and Configuration Guide* | Deployment procedures for a Red Hat Enterprise Linux OpenStack Platform cloud; procedures for both a manual and Foreman installation are included. Also included are brief procedures for validating and monitoring the installation, along with an architectural overview of Red Hat Enterprise Linux OpenStack Platform. |
| *Release Notes* (this document) | Information about the current release, including notes about technology previews, recommended practices, and known issues. |

# Chapter 2. Release Notes

These release notes highlight technology preview items, recommended practices, known issues, and deprecated functionality to be taken into consideration when deploying this release of Red Hat OpenStack.

Notes for updates released during the support lifecycle of this Red Hat OpenStack release will appear in the advisory text associated with each update or the *Red Hat Enterprise Linux OpenStack Platform Technical Notes*. This document is available from the following page:

https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform

## 2.1. Enhancements

This release of Red Hat Enterprise Linux OpenStack Platform features the following enhancements:

**BZ#842136**

```
With this update, migration operations will not attempt to copy
data when the specified source and destination are the same
filesystem location. This is due to the consideration that
recopying the data would be inefficient and may risk corrupting
the file.
```

**BZ#890123**

```
The 'System Info' panel in the Administrator dashboard now has two
additional tabs displaying service status.
The 'Compute Services' tab provides additional information on the
Compute services. Details include host and current status for
nova-conductor, nova-compute, and nova-cert.
The 'Network Agents' tab is available when Networking is enabled,
and provides data on Network Agents such as L3 and DHCP.
```

**BZ#890161**

```
Instances can now be started from the Instances list page.
Previously, there was no clear method to start an instance in a
shutdown state without using workarounds such as 'Reboot'.
This enhancement addresses this by adding a new 'Start' option in
the Instances list page for shutdown instances.
```

**BZ#890176**

```
With this enhancement, the nova client more consistently uses the
term 'server' in help text.
Previously, the terms 'server' and 'instance' were used
interchangeably.
```

**BZ#890211**

```
Compute is now able to access images directly from local and
locally mounted filesystems. Previously, only 'file://' URLs were
```

directly accessible.
This enhancement adds the concept of download plug-ins. Protocol modules enable Compute to analyse the URL scheme and load the corresponding plugin. The instance is able to directly download the image without routing through Image Service first. The download is reverted to Image Service if any failures occur in the process.

**BZ#890514**

Dashboard now makes it simpler to soft reboot multiple instances concurrently. Previously, selecting multiple instances for soft reboot from the Instance list page wasn't possible; each reboot needed to be performed individually.
This enhancement adds a new 'Soft Reboot Instances' button on the Instances list page when selecting multiple instances.

**BZ#894778**

Metering resource usage graphs are now available in Dashboard. A new 'Resource Usage' panel is accessible in the Admin dashboard, enabling administrators to query and visualise Metering telemetry data.

**BZ#894779**

Orchestration Stacks management is now available in Dashboard. This enhancement adds a new 'Stacks' panel under 'Orchestration' in the Project dashboard.

**BZ#894782**

This update adds a host aggregate view to Dashboard. A read-only 'Host Aggregates' tab is now available in the Administrator's System Info panel.

**BZ#907529**

Packstack can now configure Block Storage for remote NFS and GlusterFS shares, with the corresponding driver configured interactively or via the answer file. Currently supported backends are 'lvm', 'nfs', and 'gluster'.

**BZ#916301**

Packstack now automatically configures Block Storage for NFS when a full 'ip:/path/to/share' remote NFS path is provided. Consequently, Block Storage is configured to use NFS instead of the local LVM. Prior to this enhancement, manual configuration of Block Storage for NFS will have been required after Packstack installation.

**BZ#919607**

Packstack now automatically configures Block Storage for GlusterFS when a full 'ip:/volume/name' remote GlusterFS volume path is provided. Consequently, Block Storage is configured to use GlusterFS instead of the local LVM. Prior to this enhancement, manual configuration of Block Storage for GlusterFS would have been required after Packstack installation.

**BZ#919816**

The Image Service CLI now displays a progress bar for image upload and download activity. Prior to this enhancement, progress feedback of Image Service CLI activity was not available.

**BZ#920982**

The glance-registry service is now optional for Image Service API v2 deployment. Excluding the glance-registry service reduces the latency of Image Service database activity, and reduces code duplication throughout Image Service. The Image Service registry can still be deployed for legacy support; or as a secure method for accessing the database, rather than the public glance-api service.

**BZ#962818**

Red Hat Enterprise Linux OpenStack Platform allows you to change the flavor size of a running instance by using the 'Resize Instance' option available in the 'More' dropdown list on the Project dashboard.

**BZ#967704**

The openstack-status command output now includes the status of Orchestration and Metering services.

**BZ#970283**

Volume Service services, such as DHCP, are dependent on the host group 'Neutron Networker'. A dedicated host is recommended for the 'Neutron Networker' host group.

**BZ#970996**

Project quotas for networking can now be viewed and managed in Dashboard. This enhancement requires Networking be enabled; with a backend plugin that supports quotas.

**BZ#975941**

With this update, Block Storage volume snapshots can now be created for GlusterFS Block Storage volumes attached to Compute VMs.

**BZ#975943**

This update enables Compute to snapshot a GlusterFS volume on behalf of Block Storage. The snapshot operation is initiated by the user through Block Storage, using the same method as other volume types. Consequently, Block Storage involves Compute to complete aspects of the snapshot process.

**BZ#976272**

With this update, Dashboard users are automatically logged out of a period of inactivity. This period is set to 30 minutes by default.

**BZ#976287**

Hypervisor information is now available in Dashboard's Admin dashboard. A new 'Hypervisor' panel provides information on existing Compute nodes, including Host, Type, and vCPU.

**BZ#976449**

Instances can now be created in preferred cells. Previously, the cell scheduler would select a destination at random from known cells. Filters and weights can now be defined by administrators to direct builds to a particular cell. The keys scheduler_filter_classes and scheduler_filter_classes are configured under the cells group in nova.conf.

The new weights and filters available:
ram_by_instance_type: Select cells with the most capacity for the instance type being requested.
weight_offset: Modifies the DB to weight a particular cell. Also useful for disabling a cell.
mute_child: Assigns negative weight to cells that have not sent an update to their parent cell.
target_cell: Specifies a scheduler.

**BZ#976455**

Custom quotas defined for tenants and users can now be reverted back to the default. The Compute client recognises the new syntax:
nova quota-delete --tenant <tenant-id>
nova quota-delete --user <user-id>

**BZ#976456**

Previously, it was not possible to specify which address should be associated with the instance's floating IP address.
With this enhancement, users are able to specify the fixed IP address that a floating IP should be mapped to.

**BZ#976457**

With this update, live migration of VMs is now supported between hosts in the same cell. Users are able to live migrate VMs between hosts when using a cells-based deployment.

**BZ#976459**

As a new feature, the MAC addresses associated with an instance's allocated IP addresses are now returned as part of the instance information.

This feature allows the relevant MAC addresses to be more readily available.

MAC addresses are now present in the instance detail information returned by OpenStack Compute.

**BZ#976461**

A new feature makes 'IP attribute' available to all tenants in server search.

Previous releases allowed only admin tenants to search by IP attribute. It is a valuable query criteria when managing large numbers of VMs and has been made available to all tenants.

Now, any tenant can use the IP attribute in server searches.

**BZ#976905**

With this update, clones can now be created from GlusterFS Block Storage volumes and volume snapshots.

**BZ#978490**

Copmute is now able to use Block Storage mount options when mounting GlusterFS and NFS shares. This update enables Block Storage configuration options such as 'backupvolfile-server' to be passed to Compute.
Consequently, Compute mounts NFS and GlusterFS shares with the same arguments Block Storage uses to mount them.

**BZ#978502**

Free RAM information for a given cell was previously stored within StateManager, but was not accessible to administrators. With this update, free RAM information can now be accessed from the Compute API.
Consequently, administrators are able to perform more accurate RAM capacity planning, thereby enabling more efficient use of their OpenStack deployment.

**BZ#978514**

Support for PCI passthrough has been added to Compute, exposing a host's physical PCI devices directly to an instance. Consequently, instances can be given access to a host's physical PCI devices.

**BZ#978519**

Volumes attached to a specific instance can now be listed using the new API extension 'os-extended-volumes'. Previously, all volumes had to be iterated to identify those attached to a specific instance. The new extension provides all attached volumes together with the instance details.

**BZ#978522**

Feature:

Users may now stop servers they are not currently using but still retain them in their list of servers. This is referred to as shelving. A shelved server can be restarted using its same data, configuration, and identity at a point in the future. While a server is shelved, hypervisors are able to reclaim the resources that were used by that server and use it for running other servers if needed.

Reason:

Users would like the ability to stop servers when they do not need them, but retain it in their list of servers and retain all data and resources associated with it. If an instance is stopped for an extended period of time, an operator may wish to move that instance off of the hypervisor in order to minimize resource usage.

**BZ#978532**

Quota usage for specific tenants is now available to administrators through the API extension os-used-limits-for-admin. The Compute client recognises the new syntax:
nova absolute-limits --tenant <tenant id>

**BZ#978537**

Default quotas can now be defined from the command line and dashboard. The new defaults are stored in the database under the class 'default', and will override any defaults specified in nova.conf.
A nova restart is not required when the defaults are defined with the nova client:
nova client nova quota-class-update default <key> <value>

**BZ#978538**

Quotas can now be applied to a user within a tenant. The Compute client recognises new syntax for creating, updating, and

displaying user quotas. The quota allowance of a user cannot exceed the total granted to the tenant.
Show user quotas: nova quota-show --tenant <tenant id> --user <user id>
Update/Create user quotas: nova quota-update <tenant id> --user <user id>

**BZ#978543**

A feature has been added to propagate the hostname information to OpenStack Networking when allocating the network.

Previously, where required, OpenStack Networking would generate hostnames that were a "hash" of available information. Using a different hostname than the information provided prevented the potential of using that information to provide typical network services (e.g. DNS)

Now, hostname information for an instance is used whenever relevant in OpenStack Networking

**BZ#978565**

Feature: Heat environment

Reason: This provides a mechanism to do the following:
1) Place parameters into a file instead of the command line.
2) Map TemplateResources to named Resource types.

**BZ#978574**

With this update, when a user gets logged out and redirected to the login page, for instance, due to changing their password or the session timing out, a notification message as to why there were redirected to the login page is displayed.

**BZ#979092**

OpenStack plugins are now available for the sosreport tool. The additional plugins are available in the 'sos-plugins-openstack' package, and gather support data from OpenStack deployments.

**BZ#979138**

In the OpenStack Image service, when a direct URL is returned to a client it is helpful to the client to have additional information about that URL. For example, with a file:// URL the client may need to know the NFS host that is exporting it, the mount point, and FS type used.

This feature requests that each storage system has a means to return direct URL specific meta-data to the client when direct_url is enabled. Although all stores could support this feature already, only the filesystem one does.

It does that by reading the metadata information from an external path, configurable through a configuration parameter `filesystem_store_metadata_file` in glance-api.conf

**BZ#979142**

Glance works as a registry for images. Each record is considered an immutable resource that holds information about the image it represents. In previous versions only one location could be associated with one record. An image persisted in its location in Glance.

With this enhancement, it is now possible to associate one record to multiple locations. These locations can point to any enabled store. This enhancement can eventually be used to configure failover stores, which in turn would add High Availability capabilities to Image services. For now, using multiple locations for one records can be used to replicate images in different stores.

**BZ#982357**

An enhancement has been made to support advanced features of Neutron Security Groups. These groups have additional features, such as direction, IP version, etc. that could not be set previously when using the Dashboard to manage the rules.

Now, when using OpenStack Networking after updating the appropriate Dashboard setting, the improved security groups rule management is available.

The setting should be updated in /etc/openstack-dashboard/local_settings:
Set 'enable_security_group' to True in the OPENSTACK_NEUTRON_NETWORK key. (True is the package default).

**BZ#985139**

A feature has been added which provides basic quotas support for OpenStack Image service (Glance). A storage quota has been added that is applied against the sum total of a user's storage consumption, across all configured storage systems. A single quota is applied to all users via the configuration option 'total_storage_quota'.

**BZ#985454**

Quality of service (QoS) is now configurable for Block Storage volumes.  This update provides a framework in Block Storage for both front-end (Compute) and back-end (Block Storage) QoS based on volume type and user-defined specifications. This feature is managed with the Block Storage CLI 'cinder qos-*' commands.

**BZ#985457**

With this release, Red Hat Enterprise Linux OpenStack Platform supports resizing a volume, using the extend_volume parameter. This would add an API call to specifically extend the size of an existing volume that is currently in an available state.

**BZ#985458**

Red Hat OpenStack now allows the transfer of ownership of Block Storage Volumes, by providing a mechanism to let another tenant take ownership of a volume.

This can be used, for instance, when bespoke bootable volumes or volumes with large data sets are produced by a supplier, and then transferred to a customer.

The process requires that both parties agree to the operation and can securely exchange an authorization key. Once the volume is created, it can be transferred to the end-user using this process.

**BZ#985460**

Feature: Scheduler hints which allow users to pass in extra information that a Cinder scheduler can use to make decisions about volume placement.

Reason: New feature in OpenStack Havana

**BZ#985953**

A feature has been added to OpenStack Block Storage which allows migrating a block storage volume from one backend to another. This is an admin-only interface used for manually invoking volume migration. It is useful for manual tuning and scripting.

**BZ#986053**

A feature has been added to bind a token to a kerberos token. This has been added to circumvent the following scenario:

The current security model involves bearer tokens. This means, if you hold the token, then you are the specified user and have all the privileges associated with it.
If another user were to acquire that token, they too become that user with all those privileges.

Binding a token means that a user must have both a token and the associated cryptographic identity (in this case kerberos ticket) for the token to be valid.

Tokens can be optionally bound to a Kerberos ticket.

An article on token binding and configuration can be found here: https://github.com/openstack/keystone/blob/master/doc/source/configuration.rst#token-binding

**BZ#986099**

The following changes in configuration have been made as part of moving to the oslo library, and affect all OpenStack Networking (Neutron) plugins.

- the 'DATABASE' section in the plugin-specific ini files has been deprecated in favor of a 'database' section in neutron.conf.
- 'sql_' prefixed options have been deprecated in favor of options that do not include the prefix (e.g. sql_connection becomes connection).
- 'reconnect_interval' option has been deprecated in favor of 'retry_interval'.
- 'sql_dbpool_enable' option is no longer supported.
- 'sqlalchemy_pool_size' has been deprecated in favor of 'min_pool_size' and 'max_pool_size'.
- 'sqlalchemy_max_overflow' has been deprecated in favor of 'max_overflow'.
- 'sqlalchemy_pool_timeout' has been deprecated in favor of 'pool_timeout'.

The changes are backwards compatible, so the previous configuration format will still work.

**BZ#986367**

OpenStack Metering now has "transformers", which are a way of calculating a rate-of-change, or doing unit scaling conversions. Transformers are configurable and the values they produce as output can be emitted as derived meters.

This enables derived meters to be calculated from the original observations via configurable rules, for example simple scaling such as conversion from °F to °C, or more involved derivation of gauge rates from cumulative totals.

OpenStack Metering now provides a generic transformer for such conversions, that is used out-of-the-box for deriving CPU utilization percentage from cumulative CPU time, and may also be re-used for any additional cumulative-to-gauge conversions required via simple configuration.

**BZ#986378**

Feature: Evaluation of alarms based on comparison of static thresholds against statistics aggregated by Ceilometer.

Reason: This enhancement allows users to be notified when the performance of their cloud resources cross certain thresholds, and also allows automated workflows such as Heat autoscaling to be

triggered.

Result: New ceilometer-alarm-evaluator and ceilometer-alarm-notifier services are provided.

**BZ#986381**

Feature: Partitioning of alarm evaluation over a horizontally scaled out dynamic pool of workers.

Reason: This enhancement allows the evaluation workload to scale up to encompass many alarms, and also avoids a singleton evaluator becoming a single point of failure.

Result: The alarm.evaluation_service configuration option may be set to ceilometer.alarm.service.PartitionedAlarmService, in which case multiple ceilometer-alarm-evaluator service instances can be started up on different hosts. These replicas will self-organize and divide the evaluation workload among themselves via a group co-ordination protocol based on fanout RPC.

**BZ#986393**

A feature has been added in OpenStack Metering (Ceilometer) which allows the retention of alarm history in terms of lifecycle events, rule changes and state transformations.

This was required because alarms encapsulate a transient state and a snapshot of their current evaluation rule, but users also need the capability of inspecting how the alarm state and rules changed over a longer timespan, including the period after the alarm no longer exists.

Now, alarm history is configurably retained for lifecycle events, rule changes and state transformations.

**BZ#986410**

Feature: Aggregation of the states of multiple basic alarms into overarching meta-alarms.

Reason: Reducing noise from detailed monitoring, also allowing alarm-driven workflows (e.g. Heat autoscaling) to be gated by more complex logical conditions.

Result: Combination alarms may be created to combine the states of the under-pinning alarms via logical AND or OR.

**BZ#986498**

The main rootwrap implementation has been updated with new filtering features. These new features:

* permit more maintainable filter rules with less duplication, and

```
* reduce the security risk of invoking commands via rootwrap.

This update merges these rootwrap features into the OpenStack
Networking service.
```

**BZ#986500**

```
With this update, you can now specify protocol numbers when
defining security group rules. In some cases, this feature makes
the creation of security group rules more convenient.

The OpenStack Networking service still supports the use of
protocol names as normal.
```

**BZ#988358**

```
A public v2 API was extended for OpenStack Metering (Ceilometer),
which exposes create, retrieve, update and destroy operations for
alarms. This was required because Ceilometer alarms are a user-
oriented feature, reflecting the user's view of cloud resources
(as opposed to the cloud operator's view of the data center
fabric). Hence an alarm's lifecycle must be accessible to normal
and admin users.
```

**BZ#988362**

```
A public RESTful API to accept incoming sample datapoints has
been added to OpenStack Metering. This was required because it is
often more convenient for user-level agents to emit statistics to
Ceilometer via a public API, than to gain access to the AMQP bus.

Now, samples may be POSTed to the "/v2/meters/<meter_name>"
endpoint.
```

**BZ#989615**

```
Feature: It is now possible to create per-tenant VPN networks
through Openstack Networking.
```

**BZ#989626**

```
Allow disabling the default SNAT behavior on external networks.
```

**BZ#989676**

```
Feature: TemplateResources

Reason: This feature enables a user to customize an existing
Resource type or to create a new Resource type using a Template
instead of a python plugin.
```

**BZ#989681**

The Orchestration service now accepts existing Keystone tokens, which Heat can use to interact with the underlying services. By default, python-heatclient now creates a token on your behalf, which is then passed to the Heat API. Alternatively, you can specify an existing token via the '--os-auth-token' option.

Token support in Heat allows enhanced security over simple username/password authentication. This feature also reduces Keystone token generation, thereby reducing Identity service overhead.

**BZ#989684**

With this release, OpenStack provides new suspend/resume action for Orchestration service to enable a running stack to be suspended, which means all instances in the stack will be suspended. This helps users who want to create a stack, and suspend it in a fully configured state, such that it may be resumed more quickly than having to create it from scratch.

**BZ#995967**

A new tool for collecting logs, called rhos-logs-collector, has been introduced. There is a knowledge base article with more information available at https://access.redhat.com/site/solutions/472743

**BZ#996176**

Feature: Add block storage (Cinder) support.

Reason: Previously we did not install Cinder with Foreman.

Result (if any):

* Cinder API has been added to the Controller hostgroups (both "Controller (Neutron)" and "Controller (Nova Network)".

* Cinder storage (where block storage data is stored) has a separate host group "LVM Block Storage", so that the storage capacity is scalable. This host group can be deployed only on hosts where "cinder-volumes" LVM volume group exists.

**BZ#996596**

Schema properties of each available resource type are now presented through the Orchestration API. Consequently, automated Orchestration clients are able to query the properties of a resource type.

**BZ#998599**

This enhancement implements SSL encryption for the mysql and qpid services. In addition, SSL is enabled for Dashboard.

Consequently, qpid messages and mysql queries are hardened against snooping even on the internal network. SSL implementation in Dashboard assists with protecting user credentials sent from browser sessions.

**BZ#1001369**

When setting up a Django application such as Horizon, there are common issues. It is recommended to set up service horizon via SSL. Additionally one can use python-django-secure, which provides a script, to check if the currently configured solution is safe.

More information about django-secure can be found at it's online docs http://django-secure.readthedocs.org/en/latest/

**BZ#1001916**

PackStack now supports multiple hosts with the openstack-nova-network service. In line with this, the setting CONFIG_NOVA_NETWORK_HOST has been changed to CONFIG_NOVA_NETWORK_HOSTS.

**BZ#1002063**

Previously, service ports have been opened on all hosts in iptables. With this patch, service ports are opened only for required hosts.

**BZ#1002719**

The diskimage-builder tool is now included in this release. This tool can be used to building disk images, file system images, and ramdisk images for OpenStack.

The diskimage-builder tool can also automate image customization, which makes testing and instance management easier.

**BZ#1003120**

Feature: Add support for GRE networks.

**BZ#1005727**

With this release, PackStack introduces new parameter CONFIG_SWIFT_HASH which can be used to set swift-hash/swift_hash_path_suffix value in swift.conf file.

**BZ#1017144**

Previously, it was not possible to use the ML2 L2 plugin with OpenStack Networking.

This has been fixed so that the CONFIG_NEUTRON_L2_PLUGIN parameter accepts the value "ml2", and CONFIG_NEUTRON_ML2_* parameters have been introduced to enable OpenStack Networking to work with the ML2 plugin.

**BZ#1019780**

Packstack will now (optionally) install and configure OpenStack Networking LBaaS agent and plugin.

To deploy the LBaaS agent and plugin, use the --neutron-lbaas-hosts option (or CONFIG_NEUTRON_LBAAS_HOSTS parameter in the PackStack answer file) to list the hosts on which the LBaaS agent should be deployed. The LBaaS plugin will be configured and the agent will be configured and enabled on each host listed in the parameter.

**BZ#1020958**

Previously, Nova would default to cold snapshots of instances. As a result, instances needed to be shutdown before the snapshot was taken.

With this enhancement, Nova now uses live snapshots by default. Instances remain powered on during snapshots, and the process is unnoticeable to the user.

**BZ#1021778**

Previously, it was not possible to run neutron with Open vSwitch using Virtual Extensible LAN (VXLAN). This update adds support for VXLAN in neutron.

**BZ#1026535**

Automatic IP address assignment for instances is now optional. This feature is managed by the host group parameter 'auto_assign_floating_ip' for Compute controller.pp and compute.pp.

**BZ#1027722**

By default, the logs of OpenStack services are now checked at the system logrotate frequency; logs are then rotated when their size reaches 10MB. This is in addition to the default logrotate frequency setting each service uses.

This update was implemented to reduce disk waste and ease log management. Previously, logs were rotated weekly with no size limit; in some cases, this resulted in the creation of excessively large log files.

**BZ#1029722**

The Compute service periodically checks for instances that have been deleted in the database but remain running on a compute node. The action to take when such instances are identified is determined by the value of the "running_deleted_instance_action" configuration key in the OpenStack Compute (/etc/nova/nova.conf) file.

The default value of the "running_deleted_instance_action" configuration key has been changed to "reap". In previous releases the default value of this configuration key was "log".

As a result of this change when running instances are discovered that have previously been deleted from the database they are now stopped and removed.

**BZ#1033135**

Dashboard users can now select a destination Availability Zone (AZ) when creating an instance. A selector has been added to Dashboard's instance creation process.

**BZ#1033243**

An error message that may occur when 'foreman_client.sh' is executed has been amended for clarity. A subsequent message will now follow indicating that the error message can be ignored under a specific circumstance.

**BZ#1033247**

Feature: Provided boolean values within the parameters section of host groups to disable specific services.

Reason: For example, if a user wants to apply the host group of nova, but does not want ceilometer or heat to start they could set a parameter named "ceilometer" to FALSE.

Result (if any): The ability to turn services on or off is presently only available in the HA host group.

**BZ#1033249**

With this update, OVS bridge name and associated IP values are no longer hardcoded assumptions. OVS bridges can now be renamed, and IP association is now optional.

**BZ#1039910**

With this release, subscription-manager support when provisioning hosts has been added to Foreman. As a result, subscription-manager can now be used by setting the following parameters globally, on a per-OS or per-host basis:

subscription_manager = true

```
subscription_manager_username
subscription_manager_password
subscription_manager_repos = comma separated list of repositories
to enable
```

**BZ#1043580**

A new feature in OpenStack Orchestration in this release is the addition of the heat-templates upstream repository and tools. This provides example templates for both WordPress and OpenShift on OpenStack.

**BZ#1045054**

In OpenStack Block Storage, a feature has been added which allows the migration of a Block Storage volume between different Block Storage backends. This allows the relocation of a storage volume to a different host (or storage backend).

**BZ#1046070**

Previously, instances connected to tenant networks gained outside connectivity by going through an SNAT by the L3 agent hosting that network's virtual router.

With this release, the ability to disable SNAT/PAT on virtual servers is added ensuring that an instance in a tenant network subnet will retain its IP address as it passes through external networks. For example, if 10.0.0.1 is an instance in the 10.0.0.0/8 tenant network, R1, a virtual router that connects the 10.0.0.0/8 subnet to the 20.0.0.0/8 public provider networks, then you can use the 'neutron router-gateway-set --disable-snat R1 public' command and any traffic from 10.0.0.1, which is forwarded out to the provider network will retain its actual source IP address of 10.0.0.1.

This can be a flexible and useful method to connect instances directly to a provider network, while retaining it in a tenant network.

**BZ#1049121**

The HA MySQL Host Group provides a mechanism to set up a HA active/passive MySQL cluster which may be pointed to as the database backend for the various OpenStack service endpoints.

To use HA/MySQL within an OpenStack deployment, you need to make sure to have your MySQL cluster up and running *before* spinning up the controller node(s), and make sure the controller Host Group has correct database IP (the virtual MySQL cluster IP) specified.

At a high level, the required steps are:

1. Edit the default HA MySQL Host Group Parameters relevant to

```
your environment.
  2. Run the puppet agent to get the MySQL cluster up and verify
that the cluster was set up with no errors.
  3. Run the puppet agent again to create the OpenStack databases
and users.

Note the above does not set up any fencing configuration. In a
production environment fencing is critical: you can either use
pacemaker commands to setup fencing, or update the default HA
mysql puppet manifests.

HA MySQL Resources:

The HA MySQL Host Group is responsible for starting Pacemaker and
configuring the following resources within a resource group: the
floating IP, shared storage for the MySQL server, and the mysql
server process. For example, run the following command:

# pcs status
...
Resource Group: mysqlgrp
    ip-192.168.200.10  (ocf::heartbeat:IPaddr2):       Started
192.168.202.11
    fs-varlibmysql     (ocf::heartbeat:Filesystem):    Started
192.168.202.11
    mysql-ostk-mysql   (ocf::heartbeat:mysql):         Started
192.168.202.11

Repository Requirements:

The cluster nodes must be subscribed to the rhel-ha-for-rhel-6-
server-rpms repository, or equivalent (before being subscribed to
the HA MySQL Host Group).

HA MySQL Host Group Parameters:

To edit the Host Group Parameters in the Foreman web UI, click
More (on the top right), then Configuration, then Host Groups.
Click HA MySQL Node. Click the tab on the right, Parameters. For
any parameter you want to override (which very well could be all
of them), click on the override button and edit the value at the
bottom of the page.

A number of the parameters (especially the IP-related parameters)
have defaults which must be changed to reflect your environment.
Please take care to ensure all the parameters in your Host Group
are correct for your setup.

* mysql_root_password, cinder_db_password, glance_db_password,
keystone_db_password, nova_db_password: the MySQL database
passwords. Note that the random values displayed were generated
when you installed foreman but you may wish to override them.

* mysql_bind_address: the address mysql listens on. Only two
values make sense here, 0.0.0.0 (to listen on all address on
whichever host mysql is running on) or the same value as
```

mysql_virtual_ip.

* mysql_clu_member_addrs: the IP addresses (as a space-separated list) internal to the cluster that pacemaker communicates on. So, if you are going to have a cluster of 3 members, the three IP addresses of the cluster members are listed here. NOTE: these IP's must already be configured and active on the cluster-hosts-to-be before they are added to this Host Group (i.e., this Host Group does not set up these IP's for you).

* mysql_resource_group_name: the name of the resource group. This Host Group adds a virtual IP, filesystem and mysql resource to the resource group named here. The default is fine.

* mysql_shared_storage_type: e.g., nfs or ext3. The type of filesystem that pacemaker is responsible for mounting to /var/lib/mysql.

* mysql_shared_storage_device: the path to the storage device. E.g. if mysql_shared_storage_type, the nfs mount point.

* mysql_virtual_ip: the virtual IP address that pacemaker will manage and the IP address that clients will use to connect to MySQL.

* mysql_virt_ip_nic: the interface (e.g., eth2) that pacemaker will attempt to bring up the virtual IP on. Note that this may be empty if the host already has an IP address active on the same subnet that the virtual IP will be brought up on.

* mysql_virt_ip_cidr_mask: the subnet mask mysql_virtual_ip lives on (e.g., 16 or 24).

**BZ#1049633**

With this update, two new parameters needed by puppet-neutron for ovs agent/plugin support for vxlan have been included. Thus, adding the ability to configure VXLAN for OpenStack Networking. This update also opens the UDP port on the firewall on the OpenStack Networking and Compute nodes.

**BZ#1056055**

The default behavior most customers expect for small to medium installations is to run cinder-volume on the controller node. Packstack does this but a Foreman deployment only allows a dedicated storage backend backed by iSCSI or Red Hat Storage.

With this update, you can deploy cinder-volume on the Compute or OpenStack Networking controller node, if cinder_backend_gluster is set to false. It will detect a volume group named cinder-volumes (backed by a loopback device) and share it via the target deamon (tgtd). You must manually create the volume group, which is also the case with the cinder_backend_iscsi parameter for the LVM backend storage group.

**BZ#1056058**

The default behavior most customers expect for small to medium installations is to run cinder-volume on the controller node. Packstack does this but a Foreman deployment only allows a dedicated storage backend backed by iSCSI or Red Hat Storage.

With this update, you can deploy cinder-volume on the Compute or OpenStack Networking controller node, if cinder_backend_iscsi is set to false. It will detect a volume group named cinder-volumes (backed by an iSCSI target) and share it via the target deamon (tgtd). You must manually create the volume group, which is also the case with the cinder_backend_iscsi parameter for the LVM backend storage group.

**BZ#1056108**

In the OpenStack Block Storage service, an enhancement which allows the migration of Block Storage volume between different Block Storage backends has been added. This allows the relocation of a storage volume to a different host (or storage backend).

**BZ#1058471**

A new feature, virtio-win, is available from the Red Hat Enterprise Linux OpenStack Platform channel.

Previously, users who wished to used the virtio-win drivers were required to subscribe to the Red Hat Enterprise Linux Supplementary channel.

Now installation and configuration for virtio-win users is simplified.

**BZ#1061372**

Packstack now supports the vCenter driver. With this, Packstack can now deploy OpenStack Compute as a VM manager for a vCenter host.

This enhancement also adds the following relevant Packstack parameters:

- --os-vmware: enables VMWare features
- --vcenter-host: describes the vCenter host
- --vcenter-username: the vCenter username for the host
- --vcenter-password: the corresponding vCenter password for the supplied username
- --vcenter-cluster: the cluster where VMs will be running

**BZ#1062377**

In OpenStack Compute, new configurable methods to wipe volumes have been added. This allows more configurable tradeoffs between

performance and security.

The previous default of writing zeros over deleted volumes took a significant amount of time, and may not be needed. One can now set a global configuration setting to clear only part of a volume (to remove encryption keys for example), or to disable clearing completely.

Additionally a new 'shred' capability is available to overwrite with random data instead of zeros.

**BZ#1062670**

Foreman now installs the 'tuned' service and its related Virtual_host profile to Compute nodes. This service optimizes Compute nodes to improve the performance of hosted guests.

**BZ#1062699**

Feature: HA All In One Controller host group should allow setting Block Storage backend mount options.

Reason: This enhances Block Storage availability via RHS.

Result: Gluster shares for Block Storage are specified using "glusterfs_shares" parameter on quickstack::pacemaker::cinder class. The value should be array of Gluster shares, and can contain mount options.

Example:
["glusterserver1:/cinder -o backupvolfile-server=glusterserver2"]

**BZ#1062701**

Previously, the ability to configure Compute Network to use other types of network manager, specifically the VLAN manager were not included with OpenStack Foreman Installer.

With this release, this enhancement has been added. As a result, Compute Network is now fully configurable, so anything the upstream puppet code allows can be done using the Foreman hostgroup.

**BZ#1064050**

In the HA-all-in-one controller, file access for glance to shared storage may be specified by the parameters: backend (must be 'file'), pcmk_fs_manage (must be 'true' for pacemaker to manage the filesystem), pcmk_fs_device (the shared storage device), pcmk_fs_options (any needed mount options).

For cinder, the relevant options are: volume_backend ('glusterfs' or 'nfs'), glusterfs_shares (if using gluster), nfs_shares (if using nfs), and nfs_mount_options (if using nfs).

**BZ#1068885**

Feature: optionally create a keystonerc file on the controller node.

Reason: This saves the user some effort in that they do not need to manually set some environment variables to be able to use the CLI services on a controller.

Result: Create /root/keystonerc_admin if the user sets the controller parameter keystonerc=true.

**BZ#1071469**

Feature: Added the ability to obtain a debug dump of the system state for an OpenStack service based process. For example,upon sending SIGUSR1, trigger a dump of all native threads, green threads, live config and any other relevant info.

Reason: When troubleshooting production systems it is desirable to be able to trace all data base queries, web REST calls, messaging service RPC calls, and libvirt API calls associated with invocation of a user command or background job.

Result: You are now able to collect timing information and stack call paths for each item as a tool for analysing a series of requests to identify slow points / scalability issues.

**BZ#1073087**

Feature: Allow use of subscription-manager to register provisioned host.

Reason: The previous method using rhnreg_ks has been deprecated.

Result: To register a provisioned host using subscription-manager, make sure that the host has these parameters present and configured before provisioning:

* "subscription_manager" is set to "true"

* "subscription_manager_username" is set to subscription account user name

* "subscription_manager_password" is set to subscription account password

* "subscription_manager_repos" is a comma-separated list of repositories to enable on the host

**BZ#1075766**

A new Red Hat Storage Puppet module (GlusterFS) is now available, which provides extensive new features for advanced configurations. The puppet-gluster module is

```
from purpleidea.
This module replaces the initial puppet module, which was
temporary and offered only basic features for setting up
simple Red Hat Storage clusters and clients.
```

**BZ#1076205**

```
Feature: Makes any service optional to be installed in the HA
All-In-One-Controllers.

Reason: Not all users want to deploy all services. Whether that
means they are installed elsewhere of just not used at all, this
allows either option.

Result: There is an 'include_<service>' parameter for each service
in the HA All-In-One Controller Hostgroup, so the service can be
turned on or off.  For instance, to disable swift, you would set
the $include_swift parameter to false.
```

**BZ#1077818**

```
Feature: Added a PROVISIONING_INTERFACE environment variable to
allow users to specify something other than the inferred default.

Reason: Previously, the provisioning interface was assumed to be
the secondary interface on the foreman server. This was a bit
inflexible and made things difficult if the server had more than 3
interfaces.

Result: Users can now specify the above environment variable to
use the interface of their choice for provisioning.
```

**BZ#1082811**

```
Feature: Added Identity component to HA All In One Controller
host group.

Result: HA All In One Controller host group can be deployed
including the Identity component. Most notable parameters on
quickstack::pacemaker::keystone class:

* admin_email - the email of the admin user that will be added to
keystone.
* admin_password - the password of the above username used for
authentication.
* admin_tenant - the tenant of the above username used for
authentication.
* admin_token - in case the user already has an authentication
token, this is the place to put it.
* db_name - the database name.
* db_user - the username to log into the database.
* db_password - the password of the above username.
* enabled - set to 'true' if the service is required to survive
reboot, 'false' otherwise.
* keystonerc - set to 'true' if you want to create the keystonerc
```

```
file on the system it deploys the keystone service on.
* ceilometer, cinder, glance, heat, heat_cfn, nova, swift = all
true/false, set true if you want a keystone endpoint created for
the service.
```

**BZ#1083781**

```
Feature: Added Image service as one of the services managed by the
new HA-all-in-one controller.

Result: HA All In One Controller host group can be deployed
including the Glance component. Most notable parameters on
quickstack::pacemaker::glance class (these only are relevant if
the $include_glance param in  quickstack::pacemaker::params is
true): $backend, $pcmk_fs_manage, $pcmk_fs_options, $pcmk_fs_type,
$pcmk_fs_device, $pcmk_fs_dir, $pcmk_swift_is_local,
$swift_store_user, $swift_store_key, $swift_store_auth_address.
```

**BZ#1084534**

```
Feature: Added nova to HA All In One Controller host group.

Result: HA All In One Controller host group can be deployed
including the nova component. Most notable parameters on
quickstack::pacemaker::nova class (the majority of these will be
fine by default):

* auto_assign_floating_ip = 'true', for VMs, assign IP
automatically or not
* db_name = 'nova',
* db_user = 'nova',
* default_floating_pool = 'nova', default name for dhcp pool
* force_dhcp_release = 'false', force IPs to be returned to the
dhcp pool
* verbose = 'false', log level for nova service
```

**BZ#1086344**

```
Feature: Added Block Storage to HA All In One Controller host
group.

Result: HA All In One Controller host group can be deployed
including the Block Storage component. Most notable parameters on
quickstack::pacemaker::cinder class:

* volume - boolean switch for enabling cinder-volume service

* volume_backend - which backend to use for cinder-volume service
```

**BZ#1086815**

```
This enhancement allows administrators to rename vSphere virtual
machines created by Compute.

Previously, Compute used a virtual machine's name to look it up
```

```
in vSphere, which meant that renaming a virtual machine would
make it inaccessible to Compute. However, administrators may want
to organise virtual machines in vSphere according to their own
conventions.

With this change, administrators can now safely rename a vSphere
virtual machine created by Compute. Compute now uses other
metadata to look up the virtual machine, so it will continue to
work.
```

**BZ#1086934**

```
Feature: Added Dashboard to HA All In One Controller host group.

Result: HA All In One Controller host group can be deployed
including the Dashboard component. Most notable parameters on
quickstack::pacemaker::horizon class:

* secret_key - just needs to be set to a value, if left blank,
horizon will not start
```

**BZ#1088139**

```
Feature:  Added MySQL to HA All In One Controller host group.

Result: HA All In One Controller host group can be deployed
including MySQL. Most notable parameters on
quickstack::pacemaker::mysql class (these only are relevant if the
$include_mysql param in  quickstack::pacemaker::params is true):
$mysql_root_password, $storage_device, $storage_type,
$storage_options.

In addition, other parameters from the
quickstack::pacemaker::params class are used when setting up mysql
including: $keystone_db_password, $glance_db_password,
$nova_db_password, $cinder_db_password, $heat_db_password,
$neutron_db_password, $neutron, $db_vip.
```

**BZ#1088608**

```
Feature: Added OpenStack Networking to HA All In One Controller
host group.

Result: HA All In One Controller host group can be deployed
including the OpenStack Networking component. Most notable
parameters on quickstack::pacemaker::neutron class:

* enable_tunneling = true/false, whether you need tunneling
between VMs for networking
* enabled = true/false, whether to enable the service
* external_network_bridge = 'br-ex', name of your network bridge
* ovs_bridge_uplinks = [], a mapping between a bridge name and an
interface on the host (e.g: ["br-eth1:eth1", "br-ex:eth0"]),
IMPORTANT: the name of the external bridge (br-ex) that was
defined in 'external_network_bridge' MUST be part of this mapping
```

```
* ovs_bridge_mappings = [], a mapping between the name of the VLAN
range defined above (ovs_vlan_ranges) and the bridge name that
was defined above (ovs_bridge_uplinks)
* ovs_tunnel_iface = '', name of interface to use for tunneling
* ovs_tunnel_network = '', network as seen by puppet to use for
tunneling (determines ip or nic for you, value like 192.168.2.0)
* ovs_vlan_ranges = '', a mapping between a name of a VLAN range
to an actual VLAN range (e.g: internal_vlan_range:100-150 [,
internal_vlan_range:180-200] (the ',' is for additional ranges, if
the same name is used, it adds it to the previous range) )
* ovs_tunnel_types = [],a list of available tunnel types - vxlan,
gre, etc
* tenant_network_type = 'vlan',
* tunnel_id_ranges = '1:1000',
* verbose = 'true'/'false', whether the service should have
verbose logging
```

**BZ#1088611**

```
Feature: Added Orchestration to HA All In One Controller host
group.

Result: HA All In One Controller host group can be deployed
including the Orchestration component. Most notable parameters on
quickstack::pacemaker::params class:

* heat_cfn_enabled - boolean switch for enabling heat-api-cfn
service

* heat_cloudwatch_enabled - boolean switch for enabling heat-api-
cloudwatch service
```

**BZ#1091003**

```
This enhancement adds an enhanced wizard-based installation
process for Red Hat Enterprise Linux OpenStack Platform. The
solution is based on the staypuft foreman plugin, and utilizes
existing puppet classes used by other installation methods.
This was added due to a need for a UI-based installation; all
previous installation methods are still present.
```

**BZ#1093055**

```
This feature allows multiple concurrent deployments of Red Hat
Enterprise Linux OpenStack Platform to be managed from the
Staypuft UI.
```

**BZ#1093349**

```
RHOS 4.0 will support multiple, active instances of the Neutron
API service. This will allow higher scalability as well as a
limited form of high availability. RHOS 4.0 will use a haproxy
configuration, while RHOS 5.0 will use the HA Neutron deployment
via the Foreman Staypuft installer.
```

**BZ#1095752**

Previously, the version of Orchestration (heat) in Red Hat Enterprise Linux OpenStack Platform 4 did not include the "host_routes" property of the OS::Neutron::Subnet resource that was added in later releases of Orchestration.
This change adds support for this property, which allows host routes to be specified for a subnet.

## 2.2. Technology Preview

The items listed in this section are provided as Technology Previews. For further information on the scope of Technology Preview status, and the associated support implications, refer to https://access.redhat.com/support/offerings/techpreview/.

**BZ#989638**

OpenStack Networking now provides VPN as a Service capabilities. This functionality is offered as a Technology Preview.

**BZ#1049122**

Added qpid as one of the services managed by the new HA-all-in-one controller.

**BZ#1063334**

The ability to deploy OpenStack Telemetry (ceilometer) in High Availability Mode is now provided in this release as a Technology Preview.

**BZ#1075818**

This update includes new Red Hat Enterprise Linux OpenStack Platform Installer host groups that deploy All-In-One (All Services) controllers with High Availability (HA) Components configured. Examples of such components are Pacemaker and HAProxy. Pacemaker manages the services, which include haproxy, a/p MySQL and all OpenStack control/API services in a single cluster.

These are built to allow the user to disable services at will (for example, if identical services are running outside the cluster). Components can then be reused for customized, more focused hostgroups if the user wishes to create multiple clusters dedicated to different services.

These new host groups are provided as a Technology Preview, and are included in openstack-foreman-installer-1.0.12-1.el6ost.noarch.rpm.

**BZ#1101812**

This update includes Staypuft as a Technology Preview. Staypuft

```
provides the Red Hat Enterprise Linux OpenStack Platform Installer
with an improved user interface and integrated functionality for
easier deployment of OpenStack.

Staypuft is provided by the following packages:

    foreman-installer-staypuft-0.0.13-2.el6ost.noarch.rpm
    ruby193-rubygem-staypuft-0.0.16-1.el6ost.noarch.rpm
    ruby193-rubygem-staypuft-doc-0.0.16-1.el6ost.noarch.rpm
```

## 2.3. Recommended Practices

You must take these recommended practices into account to ensure the best possible outcomes for your Red Hat OpenStack deployment:

**BZ#843302**

```
Red Hat Enterprise Linux OpenStack Platform is only supported for
use with the libvirt driver, using KVM as the hypervisor on
Compute nodes, or the VMware vCenter hypervisor driver.

Refer to https://access.redhat.com/knowledge/articles/744153 for
more information regarding the configuration of the VMware
vCenter driver.

Red Hat is unable to provide support for other Compute
virtualization drivers including the deprecated VMware "direct-
to-ESX" hypervisor, and non-KVM libvirt hypervisors.
```

**BZ#975014**

```
In order for Nova's resize command to work when using the libvirt
driver and attempting to resize between nodes (the default resize
method), the Nova users on the compute nodes must have permission
to perform passwordless SSH to the other compute nodes.

To set this up, generate SSH keys for the Nova user on each
compute node, and then add the generated keys from the other
compute nodes to the ~/authorized_keys file for the Nova user on
each compute node.
```

**BZ#976116**

```
It is recommended that systems used to host OpenStack API
endpoints are assigned fixed IP addresses or fully qualified
domain names.

Hosting of OpenStack API endpoints on systems that have IP
addresses dynamically assigned by a DHCP server results in a loss
of service if or when the assigned address changes. When this
occurs the endpoint definitions stored in the database of the
Identity service must be updated manually.
```

**BZ#1004811**

```
Block Storage configurations previously using the setting:
volume_driver=cinder.volume.drivers.lvm.ThinLVMVolumeDriver

Should migrate to settings:
volume_driver=cinder.volume.drivers.lvm.LVMISCSIDriver
lvm_type=lvm

The ThinLVMVolumeDriver alias for the volume driver will be
removed in a future release.
```

**BZ#1008668**

```
The OpenStack Image service allows images to be shared between
projects. In certain older API versions, images can be shared
without the consumer project's approval. This allows potentially
malicious images to show up in a project's image list.

In the OpenStack 4.0 release, the OpenStack Image service supports
the Image Service API v2.1 or later. This version enforces a two-
step sharing process which explicitly requires a consumer to
accept an image, thus preventing potentially malicious images
being used without the consumer's knowledge.

However, support is still provided for Image Service API v1, which
allows image sharing between projects without consumer project
approval. It is recommended to disable v1 of the Image Service
API if possible. This can be done by setting the following
directive in the glance-api.conf configuration file:

enable_v1_api = False

With Image Service API v1 disabled, OpenStack Image service API
v2 will be used which will prevent potentially malicious images
being used.

If the glance api v1 is disabled, it will be necessary to
explicitly point to the v2 from glanceclient by using '--os-
image-api-version 2'.
```

**BZ#1059526**

```
With this release, new templates are available for the following:
* Autoscaling resources with OpenStack Networking load balancer.
* For deploying OpenShift Origin on CentOS with separated node
and broker roles.

This bug fixes the following issues:
* OpenShift security group enhancements for 'mcollective'
communication and node 'waitcondition' fix for Fedora 19.
* Added DIB_REG_TYPE variable reference for OpenShift Enterprise
based 'yum' install.
* Removed restrictive security group egress rules for OpenShift
```

```
Enterprise OpenStack Networking highly available templates.

Also includes a security update to the 'yum' repo files. The 'yum'
repo file is updated to use sslverify, gpgcheck, and https.
```

**BZ#1102468**

```
Some packages in the Red Hat OpenStack software repositories
conflict with packages provided by the Extra Packages for
Enterprise Linux (EPEL) software repositories.

The use of Red Hat OpenStack on systems with the EPEL software
repositories enabled is unsupported.
```

## 2.4. Known Issues

These known issues exist in Red Hat OpenStack at this time:

**BZ#889370**

```
The identity server's token database table grows unconditionally
over time as new tokens are generated. Expired tokens are never
automatically deleted.

To clear the token table an administrator must backup all expired
tokens for audit purposes using an SQL select statement. Once the
tokens have been backed up the administrator must manually delete
the records from the database table.
```

**BZ#966094**

```
Dashboard SSL configuration is not applied when specified in the
Packstack answer file. This behaviour is due to Nagios deleting
Apache's SSL configuration.
The workaround is to disable Nagios at installation time:
"packstack --os-horizon-ssl=y --nagios-install=n --allinone"
```

**BZ#978503**

```
Compute is able to use libgfapi to connect to Block Storage
GlusterFS volumes. However, a known issue exists where
libvirt/Compute does not support Block Storage snapshots with this
feature. This means that:
* You can:
  --Use the Block Storage GlusterFS plugin to set Red Hat Storage
as the storage provider for OpenStack Block Storage.
  --Create Block Storage volumes in the above setup and attach
those volumes to a Compute instance using FUSE.
  --Create Block Storage snapshots and clones of these Block
Storage volumes.
* You CANNOT:
```

```
  --Attach those Block Storage volumes to a Compute instance via
QEMU Gluster Blockdriver libgfapi.
  --Create snapshots of a Block Storage volume that is attached to
a Compute instance via libgfapi when using the GlusterFS plugin.
```

**BZ#979204**

```
OpenStack Networking (Neutron) is not compatible with 'simple IP
management' in the Dashboard (Horizon): OpenStack Networking does
not have a concept of a 'default' floating IP pool, additionally
an OpenStack Networking floating IP can be associated with each
instance's VIF, which would also require first checking whether an
instance only has one VIF in order to enable simple association
support.

As a result, despite enabling 'simple_ip_management' in the
Dashboard settings, the user still gets asked to select a floating
IP pool when associating an IP.

There is currently no workaround for OpenStack Networking. Simple
IP management is only compatible with nova-network at this time.
```

**BZ#1015625**

```
The SELinux policy boolean 'virt_use_fusefs' is set to OFF by
default in Red Hat Enterprise Linux. As a result, GlusterFS
volumes cannot be attached via PackStack during or after
installation by OpenStack.

Workaround: Set 'virt_use_fusefs=1' using setsebool, as follows:

  setsebool -P virt_use_fusefs=1

After this setting, SELinux will allow GlusterFS attachment, and
thus GlusterFS volumes can be utilized by OpenStack.
```

**BZ#1016806**

```
Previously, the Block Storage service did not check first if it
had the required permissions to write to a GlusterFS share before
deleting a snapshot. As a result, if the Block Storage service did
not have write permissions to a GlusterFS share, any attempts to
delete a snapshot on the share would fail. No indication would be
given to the user of why the attempt failed, and the
volume/snapshot data could be left in an inconsistent state.

With this fix, the Block Storage service now checks if it has
write permissions to a GlusterFS share before deleting a snapshot.
Any attempt to delete a snapshot would fail with the correct
notification (before any data is modified) if the Block Storage
service does not have write permissions to the share.
```

**BZ#1017280**

```
Currently, puppet does not support deployment of the ML2 Neutron
```

plugin. The ML2 plugin can be used in manual configurations, or by deploying initially with the Open vSwitch plugin using PackStack, and then converting the installation to use ML2.

This last method is described on the RDO pages at http://openstack.redhat.com/Modular_Layer_2_%28ML2%29_Plugin

**BZ#1017281**

Foreman does not support deployment of the ML2 Networking plug-in. The ML2 plug-in can be implemented in manual deployments, or by initially deploying the Open vSwitch plugin using Foreman, and then converting the installation to use ML2.
Refer to the RDO documentation for further information on the conversion process:
http://openstack.redhat.com/Modular_Layer_2_%28ML2%29_Plugin

**BZ#1017305**

Network-based file systems are susceptible to timeout errors, usually related to the particular setups in which they are deployed (for example, network topology or switches).
If a network file system is used as a backend for Block Storage (cinder), it must be properly tuned to its specific purpose according best practices recommendations.

**BZ#1028678**

Horizon is built on top of Django. Django 1.5 makes the ALLOWED_HOSTS setting mandatory as a security measure. (see https://docs.djangoproject.com/en/1.5/ref/settings/#allowed-hosts)

Also, recent changes in PackStack cause this variable to be set when using Django 1.4.

The consequence is that the Horizon Dashboard only becomes accessible when using one of the host names explicitly defined in ALLOWED_HOSTS. Otherwise, an "error 500" occurs.

Workaround: Update ALLOWED_HOSTS in /etc/openstack-dashboard/local_settings to include the hostname(s) you wish to use for the Dashboard. Then restart httpd.

The dashboard will now work when accessing it using the desired hostname(s).

**BZ#1033213**

The foreman-installer prints a known default user name and password to the console. As a result, because openstack-foreman-installer makes use of foreman-installer, a default user name and password are used, which get printed to the console when running openstack-foreman-installer.

Workaround: You must change the password right after openstack-foreman-installer finishes (the installer prints a link to a page where the password can be changed).

This replaces the password with an new (hidden) one, and anyone attempting to use the displayed password will not have access.

**BZ#1033260**

When “generic receive offload” (GRO) is enabled while using GRE or VXLAN tunneling, inbound bandwidth available to instances from an external network using a OpenStack Networking router is extremely low.

Workaround: Disable GRO offloading on the network node where the l3-agent runs by adding the following line to /etc/sysconfig/network-scripts/ifcfg-ethX:

ETHTOOL_OPTS="-K ethX gro off"

where ethX is the network interface device used for the external network. Either reboot or run "ifdown ethX; ifup ethX" for the setting to take effect.

This will provide more symmetric bandwidth and faster inbound data flow.

**BZ#1040610**

Currently, a missing firewall rule for ceilometer-api causes ceilometer-api to only be accessible locally on a controller, not from other machines.

Workaround: Open port 8777 in the firewall on the controller node(s).

This will make ceilometer-api accessible from other machines.

**BZ#1041560**

Currently, PackStack does not allow all hosts to access keystone. As a result, remote callers of various API services are unable to obtain a new token, preventing use of these API services from remote hosts.

As a workaround, execute the following commands on the controller host:

```
  $ INDEX=$(sudo iptables -L | grep -A 20 'INPUT.*policy ACCEPT'
| grep -- -- | grep -n keystone | cut -f1 -d:)
  $ sudo iptables -I INPUT $INDEX -p tcp --dport 35357 -j ACCEPT
  $ sudo service iptables save
```

After doing this the remote callers of API services work correctly.

**BZ#1051047**

When you shut down a neutron-l3-agent (or it dies) and you start another neutron-l3-agent in a different node, OpenStack Networking will not reschedule virtual routers from an L3 agent to the second one. The routing or metadata remain tied to the initial L3 agent ID. As a result, you cannot have an HA environment when you have several nodes with L3 agents, with different IDs either in Active/Active or Active/Passive states.

Workaround:
You can use the 'host=' field in the agent configuration file for both L3 agents to keep the same logical ID towards neutron-server. Two hosts should never run the neutron-l3-agent at the same time with the same 'host=' parameter. And, when one L3 agent is brought down (service stop) the 'neutron-netns-cleanup --forced' script should be used to clean any namespaces and running settings left by the neutron-l3-agent.

Using this workaround, you can have virtual routers rescheduled to a different neutron-l3-agent, as long as they have the same 'host=' logical ID. When you use neutron agent-list, the host field of the neutron-l3-agent will match the 'host=' field from configuration regardless of the actual agent hostname.

**BZ#1052336**

For GlusterFS to accept libgfapi connections, "allow-insecure ports" must be set.

As a result, when accessing a cinder volume from OpenStack node, it may fail with error 0-glusterd: Request received from non-privileg ed port. Failing request.

Perform the following to avoid this issue:

Set the following volume option:
1. # volume set volName server.allow-insecure on
2. Add the following line in /etc/glusterfs/glusterd.vol: file option rpc-auth-allow-insecure on
3. Restart the glusterd service.

With this fix, GlusterFS will be able to accept libgfapi connections.

**BZ#1056890**

When upgrading from Red Hat Enterprise Linux OpenStack Platform version 3 to version 4, if the LoadBalancer plugin is used, there is an attempt to drop the servicedefinitions and servicetypes tables. These tables do not drop if quantum-server in version 3 was started prior to running the command "quantum-db-manage upgrade head".

As a result, dropping tables fails because they do not exist.

This has been fixed by first checking whether the tables exist. As a result, the upgrade from version 3 to version 4 succeeds even if LoadBalancerPlugin is in use.

There are three known issues when upgrading version 3 to version 4 with LoadBalancer. They will be fixed in a future release.

1) Issue one is with respect to re-creating table 'vips'. This will be fixed in a future release. To work around this issue run the command:

neutron-db-manage --config-file /etc/neutron/neutron.conf --config-file /etc/neutron/plugin.ini stamp f489cf14a79c

prior to the database upgrade.

2) The workaround for the second issue is renaming device_driver path in /etc/neutron/lbaas-agent.ini from

quantum.plugins.services.agent_loadbalancer.drivers.haproxy.namespace_driver.HaproxyNSDriver

to

neutron.services.loadbalancer.drivers.haproxy.namespace_driver.HaproxyNSDriver

3) To solve the third issue, the following command must be run in the mysql command-line in neutron database:

INSERT INTO providerresourceassociations (resource_id, provider_name) SELECT id, 'haproxy' FROM pools;

**BZ#1078999**

This enhancement enables mysqld performance improvement if users add the following configuration options to the /etc/my.cnf file:
innodb_buffer_pool_size = (10-20% of availble memory)
innodb_flush_method = O_DIRECT
innodb_file_per_table

These changes are expected to be implemented in the next release.

**BZ#1102466**

It is recommended that you do not run sample scripts when installing production systems. Sample scripts are for demonstration and testing only. Specifically, the openstack-demo-install script will create OpenStack Keystone accounts with default credentials.

**BZ#1102481**

Glance does not fully support a graceful restart yet. Hence, image transfers that are still in progress will be lost when Glance services are restarted. This will occur when updating the openstack-glance package.

The workaround to avoid losing images is to wait for image transfers that are in progress to complete, before updating the openstack-glance package or restarting Glance services.

If there are no image transfers in progress during installation of a new version of the openstack-glance package or during a restart of the Glance services, then this problem will not occur.

**BZ#1102484**

The limit set for Nova processes may be exceeded in very large deployments. Then a problem may occur where you get AVC denials for sys_resource and sys_admin while running Nova. For example:

```
avc:  denied  { sys_admin } for  pid=16497 comm="iptables-save"
capability=21  scontext=unconfined_u:system_r:iptables_t:s0
tcontext=unconfined_u:system_r:iptables_t:s0 tclass=capability
```

Due to the way process inheritance is set up on Linux, calling sudo inherits the caller's ulimit. Processes owned by the new UID are counted against the inherited ulimit. Transitioning to the iptables domain drops the ability to break the soft ulimit for number of processes, which causes iptables commands to fail in certain cases. Currently the limit to the number of processes is set to 2048 for the Nova user.

While this limit should work for most installations, very large deployments may need a workaround. The workaround is to increase the limit by editing the /etc/security/limits.d/91-nova.conf file. For example, change:

```
nova     soft    nproc     2048
```

to:

```
nova     soft    nproc     4096
```

**BZ#1112632**

When using NFS shared storage for Compute instance storage, Red Hat recommends that you mount the share with the noac or the lookupcache=none option to prevent NFS clients from caching file attributes (for details, see the NFS man page). This enables migration and resizing instances between compute hosts that use the shared storage, but with slight performance penalties.

```
In a future release of RHOS, this requirement may be removed; the
Release Notes will be updated when it is safe to use NFS shared
storage for the Compute instance store without enabling the noac
or lookupcache=none option.
```

**BZ#1148695**

```
When using GlusterFS as a Block Storage back-end, set
glusterfs_qcow2_volumes=True in /etc/cinder/cinder.conf of the
openstack-cinder-volume host before volumes are created.
```

## 2.5. Deprecated Functionality

The items listed in this section are either no longer supported or will no longer be supported in a future release.

**BZ#1047849**

```
In OpenStack Compute, low-level QPID debug log messages are no
longer shown by default. These previously appeared due to the
'level=debug' parameter set in the nova.conf file.
These messages can be re-enabled by setting 'qpid=INFO' under the
default_log_levels option in the nova.conf file.
```

# Chapter 3. Upgrading

## 3.1. Upgrade Overview

**Warning**

Red Hat does not support:

- Upgrading any Beta release of Red Hat Enterprise Linux OpenStack Platform to any supported release (for example, 3 or 4).
- Upgrading Compute Networking (nova-networking) to OpenStack Networking (neutron) in Red Hat Enterprise Linux OpenStack Platform 4. The only supported networking upgrade is between versions of OpenStack Networking (neutron) from Red Hat Enterprise Linux OpenStack Platform version 3 to version 4 .

Users who installed Red Hat Enterprise Linux OpenStack Platform 3 (Grizzly) can use these procedures to upgrade their systems to Red Hat Enterprise Linux OpenStack Platform 4 (Havana).

There are four methods to upgrade Red Hat Enterprise Linux OpenStack Platform. The following table provides a brief description of each.

**Table 3.1. Upgrade Methods**

| Method | Description |
|---|---|
| Using a parallel cloud | The easiest way to do an upgrade is to deploy a completely separate Red Hat Enterprise Linux OpenStack Platform 4 (Havana) environment, and slowly migrate resources over from the old Grizzly environment to the new one. This might be excessive for a lot of users, but it is something that may be considered as the least-intrusive alternative to the other options.<br><br>Because this method is based on a new installation, no upgrade instructions are included. For installation procedures, refer to either the *Installation and Configuration Guide* (foreman or manual procedures for larger environments) or the *Getting Started Guide* (packstack procedures for an all-in-one installation). |
| All at Once | In this method, you take down all of the OpenStack services at the same time, do the upgrade, then bring all services back up after the upgrade process is complete.<br><br>- **Pros:** This upgrade process is simple. Because everything is down, no orchestration is required. Although services will be down, VM workloads can be kept running if there are no requirements to move from one version of Red Hat Enterprise Linux to another.<br>- **Cons:** All of your services will be unavailable at the same time. In a large environment, the upgrade can result in a potentially extensive downtime while you wait for database-schema upgrades to complete. Downtime can be mitigated by proper dry-runs of your upgrade procedure on a test environment as well as scheduling a specific downtime window for the upgrade.<br><br>Section 3.2, “Method - Upgrade All at Once” |

| Method | Description |
| --- | --- |
| Service by Service | This method allows you to upgrade one service at a time.<br>- **Pros:** Rather than a single large service outage, you are able to stage outages to specific services. For example, you could have the Identity service running at the Havana release while Compute runs at the Grizzly release.<br>You can schedule potentially longer upgrades, such as the Compute service upgrade in a large environment, separately from upgrades that take less time.<br>- **Cons:**You will still have an interruption to your Nova APIs and Compute nodes.<br>Section 3.3, “Method - Upgrade Service-by-Service” |
| Service by Service with Parallel Computes | This method is a variation of the service-by-service upgrade, with a change in how the Compute service is upgraded. Rather than upgrading your existing Compute environment as part of the process, you deploy new nodes running the Havana Compute services. You then wait for existing workloads on your Grizzly Compute nodes to complete (or migrate them by hand), and when a Grizzly Compute node is no longer hosting any instances, you upgrade the Compute service on that node.<br>- **Pros:** This method minimizes interruptions to your Compute service, with only a few minutes for the smaller services, and a longer migration interval for the workloads moving to newly-upgraded Compute hosts. Existing workloads can run indefinitely, and you do not need to wait for a database migration.<br>- **Cons:** Additional hardware resources are required to bring up the Havana Compute nodes.<br>Section 3.4, “Method - Upgrade Service-by-Service with Parallel Computes” |

For all methods:

- Ensure you have subscribed to the correct channels for this release on all hosts (see Section 1.3, “RHN/CDN Channels”).
- The upgrade will involve some service interruptions.
- Running instances will not be affected by the upgrade process unless you either reboot a Compute node or explicitly shut down an instance.

**Note**

- If you are upgrading from an older version of Grizzly, which required Red Hat Enterprise Linux 6.4, you will also need to upgrade to Red Hat Enterprise Linux 6.6 (required by Havana). Upgrading from to RHEL 6.6 will require you to reboot both controller and Compute nodes to get the newer kernel running. Currently, there is no way to update a kernel without rebooting the base operating system.
- If you require a RHEL update, this means that your users will experience downtime of their instances (VMs) as well. This can be partially mitigated by suspending the VMs before the update/reboot and then resuming them afterwards.

## 3.2. Method - Upgrade All at Once

To upgrade everything at once, run the following steps on all of your hosts:

1. Ensure you are subscribed to the right Red Hat Enterprise Linux OpenStack Platform 4 channels on all of your hosts (see Section 1.3, “RHN/CDN Channels”).

2. Ensure the **openstack-utils** package is installed:

```
# yum install openstack-utils
```

3. Take down all the services on all the nodes. This step depends on how your services are distributed among your nodes.

   To stop all the OpenStack services running on a host, run:

```
# openstack-service stop
```

4. Perform a complete upgrade of all packages, and then flush expired tokens in the Identity service (might decrease the time required to synchronize the database):

```
# yum upgrade
# keystone-manage token_flush
```

5. 

   Upgrade the database schema for each service:

```
# openstack-db --service serviceName --update
```

   For example:

```
# openstack-db --service keystone --update
```

   For reference purposes, the following table contains the commands run by **openstack-db**.

   **Table 3.2. openstack-db commands**

| Service | Comman |
|---|---|
| Identity (keystone) | On the Identity service host, run:<br>`# keystone-manage db_sync` |
| Block Storage (cinder) | On the Block Storage service host, run:<br>`# cinder-manage db sync` |
| Object Storage (swift) | Object Storage does not require an explicit schema upgrade. |
| Image Service (glance) | On the Image API host, run:<br>`# glance-manage db_sync` |
| Compute (nova) | On the Compute API host, run:<br>`# nova-manage db sync` |
| OpenStack Networking | On the Networking service host, run:<br>`# neutron-db-manage \`<br>`--config-file /etc/neutron/neutron.conf \`<br>`--config-file /etc/neutron/plugin.ini upgrade head`<br>**Warning:** These instructions require at least version 2013.2-9 of the **openstack-neutron** package. |

6. Review the resulting configuration files. The upgraded packages will have installed **.rpmnew** files appropriate to the Havana version of the service.

7. Upgrade the Dashboard service on the Dashboard host:

```
# yum upgrade \*horizon\* \*openstack-dashboard\*
```

8. 

   Manually configure the Dashboard configuration file: **/etc/openstack-dashboard/local_settings**.

   In general, the Havana services will run using the configuration files from your Grizzly deployment. However, because Dashboard's file was substantially changed between versions, it must be manually configured before its services will work correctly:

   a. Back up your existing **local_settings** file.

   b. Replace **local_settings** with **local_settings.rpmnew**.

   c. Update your new **local_settings** file with any necessary information from your old configuration file (for example, **SECRET_KEY** or **OPENSTACK_HOST**).

d. If you are running Django 1.5 (or later), you must ensure that there is a correctly configured **ALLOWED_HOSTS** setting in your **local_settings** file. **ALLOWED_HOSTS** contains a list of host names that can be used to contact your Dashboard service:

- If people will be accessing the Dashboard service using "http://dashboard.example.com", you would set:

```
ALLOWED_HOSTS=['dashboard.example.com']
```

- If you are running the Dashboard service on your local system, you can use:

```
ALLOWED_HOSTS=['localhost']
```

- If people will be using IP addresses instead of, or in addition to, hostnames, an example might be:

```
ALLOWED_HOSTS=['dashboard.example.com', '192.168.122.200']
```

**Note**

For more information about the **ALLOWED_HOSTS** setting, see the Django Documentation.

9. Start all OpenStack services on all nodes; on each host, run:

```
# openstack-service start
```

## 3.3. Method - Upgrade Service-by-Service

This method allows you to upgrade one service at a time. At a high level, you:

1. Ensure you are subscribed to the right Red Hat Enterprise Linux OpenStack Platform 4 channels on all of your hosts (see Section 1.3, “RHN/CDN Channels”).
2. Update and test each of the services (Section 3.3.1, “Upgrade an individual service”)
3. Do a final complete-system upgrade (Section 3.3.2, “Finalize system upgrade”)

### 3.3.1. Upgrade an individual service

The following steps provide the generic procedure for upgrading a service; upgrade order and individual service notes are contained in Table 3.3, “Service Upgrade Order and Instructions”.

1. Stop the service, using:

```
# openstack-service stop serviceName
```

For example, to stop all Compute services on a host, use:

```
# openstack-service stop nova
```

```
Stopping openstack-nova-api: [  OK  ]
Stopping openstack-nova-cert: [  OK  ]
Stopping openstack-nova-conductor: [  OK  ]
Stopping openstack-nova-consoleauth: [  OK  ]
Stopping openstack-nova-scheduler: [  OK  ]
```

2. Upgrade the packages that provide the service, using:

```
# yum upgrade \*serviceName*\
```

For example, for the Compute service, use:

```
# yum upgrade \*nova\*
```

3. Upgrade the database schema for the service, using:

```
# openstack-db --service serviceName --update
```

For example, for the Compute service, use:

```
# openstack-db --service nova --update
```

For specific commands used by **openstack-db** for each individual service, see Table 3.2, “openstack-db commands”)

4. Restart the service using:

```
# openstack-service start serviceName
```

For example, to start all Compute services on a host, use:

```
# openstack-service start nova
Starting openstack-nova-api: [  OK  ]
Starting openstack-nova-cert: [  OK  ]
Starting openstack-nova-conductor: [  OK  ]
Starting openstack-nova-consoleauth: [  OK  ]
Starting openstack-nova-scheduler: [  OK  ]
```

5. Review any new configuration files (***.rpmnew**) installed by the upgraded package.

6. Test to ensure the service is functioning properly.

The following table provides specific instructions for each service, and the order in which they should be upgraded:

**Table 3.3. Service Upgrade Order and Instructions**

| Service | Notes |
|---|---|

| Service | Notes |
|---|---|
| 1.Identity (keystone) | Because the Identity service in Grizzly never purged expired tokens, it is possible that your token table has a large number of expired entries. This can dramatically increase the time it takes to complete the database schema upgrade.<br><br>To flush expired tokens from the database and alleviate the problem, the **keystone-manage token_flush** command can be used before running the Identity database upgrade.<br><br>On your Identity server, run:<br><br>`# openstack-service stop keystone`<br>`# yum -d1 -y upgrade \*keystone\*`<br>`# keystone-manage token_flush`<br>`# openstack-db --service keystone --update`<br>`# openstack-service start keystone` |
| 2.Object Storage (swift) | On your Object Storage servers, run:<br><br>`# openstack-service stop swift`<br>`# yum -d1 -y upgrade \*swift\*`<br>`# openstack-service start swift` |
| 3.Block Storage (cinder) | On your Block Storage host, run:<br><br>`# openstack-service stop cinder`<br>`# yum -d1 -y upgrade \*cinder\*`<br>`# openstack-db --service cinder --update`<br>`# openstack-service start cinder` |
| 4.Image Service (glance) | On your Image Service server, run:<br><br>`# openstack-service stop glance`<br>`# yum -d1 -y upgrade \*glance\*`<br>`# openstack-db --service glance --update`<br>`# openstack-service start glance` |

| Service | Notes |
| --- | --- |
| 5.OpenStack Networking (quantum/neutron) | **Warning:**This can only be done if you are using 'quantum' networking in your Grizzly environment.<br><br>On your OpenStack Networking (quantum) hosts and Compute nodes, run:<br><br>`# openstack-service stop quantum`<br>`# userdel quantum`<br>`# yum upgrade \*quantum\*`<br><br>On your OpenStack Networking (now neutron) host, run:<br><br>`# openstack-db --service neutron --update`<br><br>On your OpenStack Networking host and Compute nodes, run:<br><br>`# openstack-service start neutron` |
| 6.Dashboard (horizon) | On your Dashboard host, run:<br><br>`# yum upgrade \*horizon\* \*openstack-dashboard\*`<br><br>Manually configure the **/etc/openstack-dashboard/local_settings** file (see Step 8 in Section 3.2, “Method - Upgrade All at Once”). |
| 7.Compute (nova) | On all hosts running Compute services, run:<br><br>`# openstack-service stop nova`<br>`# yum -d1 -y upgrade \*nova\*`<br><br>On your Compute API host, run:<br><br>`# openstack-db --service nova --update`<br><br>On all your hosts running Compute services, run:<br><br>`# openstack-service start nova` |

### 3.3.2. Finalize system upgrade

After completing all of your service upgrades, you must perform a complete package upgrade on all of your systems:

1. Upgrade the system:

   ```
   # yum upgrade
   ```

   This upgrades the client packages on all of your systems (for example, packages like **python-keystoneclient** or **python-glanceclient**) as well as generally ensuring that you have the appropriate versions of all supporting tools.

2. Restart the Compute service (which otherwise will encounter errors due to the upgrade of the Image client package):

```
# service openstack-nova-compute restart
```

3. If this results in a new kernel being installed on your systems, you will probably want to schedule a reboot at some point in the future in order to activate the kernel.

## 3.4. Method - Upgrade Service-by-Service with Parallel Computes

This method is a variation of the service-by-service upgrade, with a change in how the Compute service is upgraded:

1. Upgrade your system service-by-service, but stop after completing the Dashboard upgrade (do not upgrade Compute). For details, see Table 3.3, “Service Upgrade Order and Instructions”.
2. Run the final **`yum upgrade`** command on systems that are not running Compute services. See Section 3.3.2, “Finalize system upgrade”.
3. Set up a parallel Compute environment, installing a new Compute controller and Compute nodes using the Havana repositories. These systems use a configuration that is basically identical to that on your Grizzly Compute nodes, making use of the same services that are also in use by the old un-upgraded Compute deployment (for example, Identity, Image, or Block Storage). See Section 3.4.1, “Set up a parallel Compute environment”.

   Using a parallel environment allows you to slowly migrate workloads to the new Compute deployment, moving and upgrading physical hosts over to the new release as they become vacated.
4. Move instances to the Havana Compute nodes. Section 3.4.2, “Move instances to the new environment”.

### 3.4.1. Set up a parallel Compute environment

Execute the following steps to set up a new Compute environment, running Red Hat Enterprise Linux OpenStack Platform 4, that is separate from your old Grizzly environment:

1. Install Compute packages using **`yum install`** ***`packageName`***.
   - On the system acting as your Compute API server, install the following packages:
     - **`python-novaclient`**
     - **`openstack-nova-common`**
     - **`openstack-nova-conductor`**
     - **`openstack-nova-novncproxy`**
     - **`openstack-nova-api`**
     - **`python-nova`**
     - **`openstack-nova-cert`**
     - **`openstack-nova-console`**

- **openstack-nova-scheduler**

- On any system acting as one of your Compute servers, install the following:
  - **openstack-nova-common**
  - **openstack-nova-compute**
  - **python-neutron**
  - **openstack-neutron**
  - **python-neutronclient**
  - **python-novaclient**
  - **python-nova**
  - **openstack-neutron-openvswitch**
  - **bridge-utils**

2. Create a new Compute database.

   The new Havana Compute environment requires a distinct database from the one your existing Grizzly Compute environment is using. On a system where you have administrative access to your SQL server, create a new database.

   For example, to create a database with the name **nova_havana**, the MySQL commands would be:

```
mysql> create database nova_havana;
Query OK, 1 row affected (0.00 sec)
mysql> grant all on nova_havana.* to nova@'%';
Query OK, 0 rows affected (0.00 sec)
```

3. Configure the Compute service:

   a. Replace **/etc/nova** on your Havana Compute controller with the contents of **/etc/nova** from your Grizzly controller.

   b. In the **/etc/nova/nova.conf** file:

   - Update **sql_connection** to point to the database created in Step 2. If your old configuration looked like this:

```
sql_connection = mysql://nova@192.168.122.110/nova
```

     The new configuration should look like this (using the example **nova_havana** database):

```
sql_connection = mysql://nova@192.168.122.110/nova_havana
```

   - Update the **metadata_host** setting to point to your new Havana controller.
   - On your Compute nodes, make sure that the following settings all point to the address of the local Compute node:
     - **vncserver_proxyclient_address**

- **novncproxy_base_url**
- **vncserver_listen**

- Change the message topics used by Compute when communicating via the AMQP server:
  - Add the following to the **[DEFAULT]** section:

```
cert_topic=cert_havana
compute_topic=compute_havana
console_topic=console_havana
consoleauth_topic=consoleauth_havana
notifications_topic=notifications_havana
scheduler_topic=scheduler_havana
```

  - Add the following to the **[conductor]** section (you will probably have to add this section):

```
[conductor]
conductor_topic=conductor_havana
```

4. Configure the Compute nodes:
   a. Enable the following services:
      - **openvswitch**
      - **messagebus**
      - **libvirtd**

      To enable a service, run:

```
# chkconfig serviceName
# service serviceName start
```

   b. Create the Open vSwitch bridge devices that will be used by the Compute service. Assuming a standard GRE tunneling configuration, run:

```
ovs-vsctl add-br br-int
ovs-vsctl add-br br-tun
```

5. Start Compute services:
   a. Start OpenStack services on the Havanna controller:

```
# openstack-service start
```

   b. Start OpenStack services on each Havana Compute node:

```
# openstack-service start
```

   c. Verify that the new Compute service has registered itself properly with the Havana controller. On the Havana controller, run:

```
# nova-manage service list
```

Which should result in one **nova-compute** entry for each Havana Compute node, as well as the following entries:

- **nova-conductor**
- **nova-consoleauth**
- **nova-cert**
- **nova-scheduler**

6. Register your new controller with the Identity service in a separate region (register Identity service endpoints):

   a. Find the service ID for the Compute service:

```
# keystone service-get nova
 +-------------+----------------------------------+
 |   Property  |              Value               |
 +-------------+----------------------------------+
 | description |    Openstack Compute Service     |
 |      id     | befb024666424084b37a84ed5ee1143b |
 |     name    |               nova               |
 |     type    |             compute              |
 +-------------+----------------------------------+
```

   b. Create a new Compute endpoint. For example, using the name **Havana** for the new region and **192.168.122.198:8774** for the endpoint URL:

```
# keystone endpoint-create --region Havana \
--service-id befb024666424084b37a84ed5ee1143b \
--publicurl http://192.168.122.198:8774/v2/%(tenant_id)s \
--adminurl http://192.168.122.198:8774/v2/%(tenant_id)s \
--internalurl http://192.168.122.198:8774/v2/%(tenant_id)s
```

   c. To use volume attachment in the Havana environment, create a new endpoint for the Block Storage service in your new region. For example:

```
# keystone service-get cinder
 ...
 # keystone endpoint-list
 ...
 # keystone endpoint-create --region Havana \
--service-id 1a6f2343a6f14bc9b5a2c2f4e4a894ca \
--publicurl 'http://192.168.122.110:8776/v1/%(tenant_id)s'
\
--adminurl 'http://192.168.122.110:8776/v1/%(tenant_id)s' \
--internalurl 'http://192.168.122.110:8776/v1/%
(tenant_id)s'
```

   d. Verify that you can communicate with the new region. After loading appropriate Identity service credentials, run:

```
# nova --os-region-name regionName host-list
```

The command should result in a listing of your new Havana Compute hosts. For example:

```
# nova --os-region-name Havana host-list
+-------------------------------------------+----------------
+----------+
 | host_name                                 | service
| zone     |
 +-------------------------------------------+---------------
-+----------+
 | rdo-havana-nova-api-net0.default.virt     | cert
| internal |
 | rdo-havana-nova-api-net0.default.virt     | conductor
| internal |
 | rdo-havana-nova-api-net0.default.virt     | consoleauth
| internal |
 | rdo-havana-nova-api-net0.default.virt     | scheduler
| internal |
 | rdo-havana-nova-compute-net0.default.virt | compute_havana
| internal |
 +-------------------------------------------+---------------
-+----------+
```

### 3.4.2. Move instances to the new environment

1. Migrate instances from the Grizzly environment to the Havana Compute nodes. The simplest method for 'migrating' an instance is to simply stop the instance running in your Grizzly environment and deploy a new one on the Havana infrastructure.

   If re-deployment is not an option, you can move instances from your Grizzly compute nodes to your Havana compute nodes with minimal downtime using the following process:

   a. Make a snapshot of the existing instance.

   b. Delete the existing instance.

   c. Boot a new instance on the Havana Compute nodes from the snapshot.

   d. Allocate and assign any necessary floating addresses:

      - In an OpenStack Networking (neutron) environment, re-assign your previously allocated addresses as soon as the instance to which they were assigned is shut down.

      - In a Compute networking environment, you will need to create new floating IP pools identical to those in your Grizzly environment and then explicitly allocate the necessary floating IP addresses before assigning them.

        To allocate explicit addresses, use the **`nova floating-ip-bulk-create`** command.

2. When you are able to move all the instances from one of your existing Grizzly Compute nodes, you can re-deploy that node in your Havana compute environment.

3. When you have moved all your Compute nodes into the Havana environment, you can retire any remaining Grizzly Compute services.

# Revision History

**Revision 4.1-37** **Tue Nov 11 2014** **Don Domingo**
BZ#1148695: added workaround for Known Issue on GlusterFS volume creation.

**Revision 4.1-36** **Wed Oct 22 2014** **Don Domingo**
Final version for Red Hat Enterprise Linux OpenStack Platform maintenance release 4.0.5.

**Revision 4.1-34** **Tue Jun 03 2014** **Don Domingo**
BZ#1063334 - OpenStack Telemetry HA Mode is now available as Technology Preview.

**Revision 4.1-33** **Thu May 29 2014** **Don Domingo**
Final version for Red Hat Enterprise Linux OpenStack Platform maintenance release 4.0.4.

**Revision 4.1-32** **Tue Apr 1 2014** **Don Domingo**
BZ#1082149 - Corrected documentation on support status of vCenter driver

**Revision 4.1-28** **Tue Mar 25 2014** **Don Domingo**
Final version for Red Hat Enterprise Linux OpenStack Platform maintenance release 4.0.3.

**Revision 4.1-25** **Wed Mar 12 2014** **Bruce Reeler**
BZ#1074308 - Expanded on description of PackStack in 1.4 Deployment Tools.

**Revision 4.1-24** **Tue Mar 11 2014** **Martin Lopes**
Final version for Red Hat Enterprise Linux OpenStack Platform maintenance release 4.0.2.

**Revision 4.1-14** **Wed Feb 26 2014** **Bruce Reeler**
BZ#1070053 - Updated Upgrade section to mention no beta upgrades supported, not just ver4.

**Revision 4.1-13** **Wed Jan 29 2014** **Scott Radvan**
BZ#1057730 - Update list of channels to disable when using CDN.

**Revision 4.1-11** **Wed Jan 22 2014** **Summer Long**
Final version for Red Hat Enterprise Linux OpenStack Platform Maintenance Release 4.0.1.

**Revision 4.1-8** **Mon Jan 20 2014** **Summer Long**
BZ#988039 - Updated product introduction with minor edits.

**Revision 4.1-7** **Wed Dec 18 2013** **Bruce Reeler**
Final version for Red Hat Enterprise Linux OpenStack Platform 4.0.

**Revision 4.1-6** **Wed Dec 18 2013** **Bruce Reeler, Summer Long**
BZ#1033152 - Updated info on Block Storage backup using openstack-cinder-backup.
BZ#1044254 - Added admonition to not install OpenStack Networking with Foreman, which is not working yet.
BZ#1022544 - Updated info on RHEL-HA support.
BZ#988040 - Updated Release Introduction.

**Revision 4.1-5** **Wed Dec 18 2013** **Summer Long**
BZ#1024577 - SCL info removed.

**Revision 4.1-4** **Tue Dec 17 2013** **Summer Long**

BZ#988031 - Updated Upgrading instructions; removed older neutron info.

**Revision 4.1-3** **Mon Dec 16 2013** **Bruce Reeler**

BZ#1031855 - Updated link to list of updates for version 4. Other minor edits.

**Revision 4.1-2** **Sun Dec 15 2013** **Summer Long**

BZ#988031 - Included detailed upgrade instructions.
BZ#989729 - Included upgrade instructions for quantum -> neutron.

**Revision 4.1-1** **Thu Dec 12 2013** **Summer Long**

BZ#1017821 - Including warning against upgrading from Compute networking to OpenStack Networking.

**Revision 4.1-0** **Wed Dec 11 2013** **Summer Long**

BZ#988031 - Upgrade overview section included for 3->4.
BZ#1039083 - New section for RHN/CDN channels.

**Revision 3.0-6** **Wed Nov 19 2013** **Summer Long**

BZ#1021939 - Requirement for RHEL 6.5 Beta added.
BZ#1023054 - Note about other RH components added to About section.
BZ#988039 - Product intro updated.

**Revision 3.0-5** **Wed Nov 8 2013** **Bruce Reeler**

Red Hat Enterprise Linux OpenStack Platform 4 BETA. Redundant bugs removed.
BZ#988039 - Release intro updated.
BZ#1025068 - Upgrade path for Beta included.

**Revision 3.0-3** **Wed Nov 6 2013** **Bruce Reeler**

Red Hat Enterprise Linux OpenStack Platform 4 BETA. Update.

**Revision 3.0-2** **Thu Oct 31 2013** **Bruce Reeler**

Red Hat Enterprise Linux OpenStack Platform 4 BETA.

**Revision 2.0-7** **Tue Jun 25 2013** **Steve Gordon**

Red Hat OpenStack 3.0 General Availability.

**Revision 2.0-2** **Tue May 21 2013** **Steve Gordon**

Red Hat OpenStack 3.0 Preview.